Game Design and Development for Learning Physics Using the Flow Framework

Danu Pranantha[1,2(✉)], Erik van der Spek[2], Francesco Bellotti[1], Riccardo Berta[1], Alessandro De Gloria[1], and Matthias Rauterberg[2]

[1] DITEN, University of Genoa, Via Opera Pia 11A, Genoa, Italy
{franz,berta,adg}@elios.unige.it, d.pranantha@tue.nl
[2] Faculty of Industrial Design, Eindhoven University of Technology, Den Dolech 2, Eindhoven, The Netherlands
{e.d.v.d.spek,g.w.m.rauterberg}@tue.nl

Abstract. Instruction, in several knowledge domains, aims at achieving two goals: acquisition of a body of knowledge and of problem solving skills in the field. In physics, this requires students to connect physical phenomena, physics principles, and physics symbols. This can be learned on paper, but interactive tools may increase the learner's ability to contextualize the problem. Computer simulations provide students with graphical models that join phenomena and principles in physics. However, a minimally guided approach may make learning difficult, since it overburdens the working memory. In particular, for developing problem solving skills, students need to be guided and exercise with a variety of physics problems. Intelligent tutoring systems (ITS) can be a useful tool to fill this gap. Thus, we have developed a physics game to support inquiry learning and retrieval practicing using simulation and knowledge based tutorship (QTut), and implemented as a puzzle game that uses driving questions to encourage students to explore the simulation. To address scalability and reusability, the game features different difficulty levels atop of a customizable format. This allows us to explore in-game adaptivity, exploiting task and user models that rely on the flow framework. User tests are being executed to evaluate the usefulness of the game.

1 Introduction

Learning physics aims at achieving two goals: the acquisition of a body of knowledge and the ability to solve quantitative problems in physics. In physics, the body of knowledge is organized into three levels: the macroscopic level corresponds to physical objects, their properties and behaviour; the microscopic level explains the macroscopic level using concepts, theories and principles of physics; and the symbolic level represents the concepts of physics as mathematical formulae [1]. Consequently, physics instructions need to advocate the connection of those levels to the students.

Lack of knowledge and/or misconceptions at the microscopic level lead students to difficulties in solving physics problems [2]. The use of concrete models,

A. De Gloria (Ed.): GALA 2014, LNCS 9221, pp. 142–151, 2015.
DOI: 10.1007/978-3-319-22960-7_14

analogies and graphics may help students to overcome difficulties. In this regard, computer simulations graphically model physical objects and unite the macroscopic, the microscopic, and the symbolic levels. This approach urges students to actively seek questions, explore the simulation, and discover knowledge based on their observations.

However, such a minimally guided approach may harm learning since it does not align with working memory limitations [3]. This, to some extent, necessitates the use of scaffolding, which is essential for inquiry learning [4]. It is also crucial for students to exercise with a variety of physics problems and to perform retrieval practices at microscopic and symbolic levels [5]. In this case, Intelligent tutoring systems (ITS) nurture students in problem solving skills. Thus, combining a physics simulation with a tutoring system in the form of serious games may provide students with a graphical tool for exploration (the macroscopic level) and a training tool for problem solving (the microscopic and the symbolic levels). Serious games have the strengths of appealing and motivating students. A meta analysis also showed that games can be more effective than traditional instructions, but only when considering working memory limitations [6].

To this end, we created an online puzzle game in physics, in particular Newtonian mechanics for the first year university students, that uses simulation to represent physical objects at the macroscopic level and a knowledge tutor (namely QTut) to explain physical phenomenon at the microscopic and the symbolic levels. The game was implemented using HTML5, JavaScript, Box2D-JS, PHP, and Ajax (Asynchronous Javascript and XML) for rich web experiences, JSON (JavaScript Object Notation) for lightweight data storage, and NLTK (natural language tool kit) for natural language processing[1]. We adhered to the flow framework [7] in designing the game to consider experiential learning [8] and followed rapid prototyping to iteratively create prototypes over short period.

2 Game System Design

2.1 Designing Educational Games and the Flow Experiences

The flow framework considers player, game artefact, and task elements for designing optimal experience (flow) in educational games (Fig. 1) [7]. The framework divides the experiential learning in games into three important phases: (a) inducing flow antecedents, i.e. factors that contribute to the flow experience and should be considered in educational game design, (b) achieving flow state, i.e. an experience where players are completely unaware of their surroundings since they are fully concentrated on solving the tasks in games, and (c) getting the outcomes of being in the flow experience in gaming (flow consequences) which include learning and exploratory behavior.

Flow antecedents include clear goals, good cognitive/immediate feedback, and autonomy for performing cognitive task and using game artefacts [9], with the addition of playability for the artefacts. The premise is, therefore, games that

[1] http://www.nltk.org/.

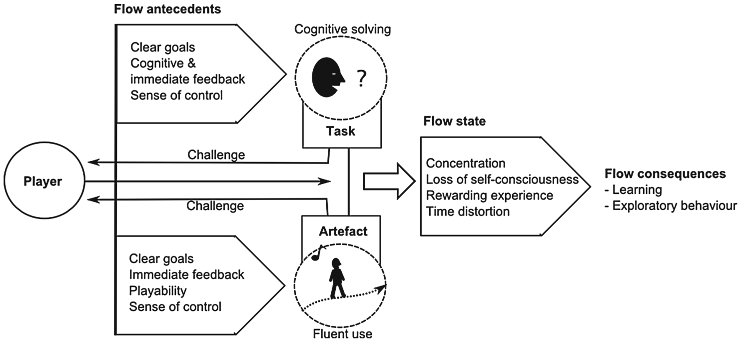


Fig. 1. The flow framework.

are well equipped with the antecedents in form of proper challenges (stimuli) are more likely to promote users reaching the flow state and, subsequently, learning. This emphasizes the importance of integrating the antecedents into game mechanics and gameplay.

2.2 Game Mechanics, Gameplay, and the Flow Antecedents

To develop the game mechanics and the gameplay, we considered two educational artefacts in the physics game: a simulation and a tutoring tool (QTut). Tasks in the game include understanding physics concepts (conceptual knowledge) and solving physics problems (procedural knowledge). To be easily grasped, we selected puzzle guessing as the primary mechanic of the game with the tutor as scaffolding. Table 1 shows the game mechanics in relation with flow antecedents.

Using the game mechanics we constructed the gameplay. All game levels are initially locked except at the base level (level 1). For simplicity, all tasks in a level have equal weights for scoring. However, each level has three most difficult tasks, with each indicated by a star. If a student answers a starred task, he will receive one star.

A level has a topic related to its preceding and succeeding levels. For instance, force and torque can be two successive levels. If a level is unrelated to its preceding, the tutor presents an introduction to denote a topic transition. A student may progress to a level (i.e., unlock a level) if he has passed its preceding level. A student completes a level if he earns at least two stars and scores above a certain threshold. During the game, a student may query the tutor about concepts, formulas, and terminologies. Moreover, relevant tools, e.g. ruler and calculator, can be used to help solving the puzzles. There is no timeout in the game but we use the timer for logging purpose.

3 Game Development

To develop the game artefacts, we started from identifying challenges in the development of the game, devising the solutions, implementing each solution as

Table 1. Game mechanics and flow antecedents of artefacts and tasks in game for learning physics.

No	Game mechanics	Flow antecedents	Elements
1	A task is defined as a puzzle where the system poses the puzzle and the player solves the puzzle in turn	Clear goal	Task
2	Game level consists of a sequence of puzzles		
3	Game level are either unlocked or locked		
4	Required metrics for unlocking a level are game score and collectibles (e.g. star) in its preceding level		
5	Both the selected and the correct answers are immediately highlighted after a user answering the puzzle	Cognitive feedback	
6	The tutor immediately provides customized text-auditory feedback		
7	Puzzles are given with increasing difficulties in each level	Sense of control	
8	Topics are interrelated for successive game levels		
9	Checkpoints are available in each game level		
10	The tutor may provide hints	Clear goal	Artefact
11	Proper symbols for representing game levels (e.g. grid lock to represent locked levels)		
12	Scaffolding using visual feedback from the simulation	Immediate feedback	
13	Scaffolding using text-auditory responses from the tutor		
14	Grouping functions of game elements into the same grid to ease navigation	Sense of control	
15	Freedom to explore the simulation (to select, to move, to rotate, to collide objects)		
16	Functionality to reset the simulation		
17	Freedom to query the tutor		
18	Providing relevant tools to solve the puzzles if necessary		
19	The use of simulation to mimic real object behavior	Playbility	
20	The use of tutor to mimic teacher		
21	Cartoonish visual graphics for the simulation		
22	Selectable cartoonish avatars for the tutor		
23	Musical background during play		

a module, and ended with integrating the modules into a complete system. We considered three challenges in developing the game system: extensibility refers to the ease to produce a variety of games for different topics, scalability means the ease to attach new modules to the system, and reusability corresponds to the use of some modules for other purposes. Therefore, to address the challenges, we created game format and knowledge based tutor, and implemented the system in a modular fashion.

3.1 Game Level and Game Format

We used game levels and created game format to allow extensibility [10]. The game level clusters learning topics into levels based on their complexity. The game format sets each game level as series of tasks -a puzzle set- drawn from the database (a JSON file). A task - or a task item- is either a closed ended question about a simulated event or an action request in the simulation area. Figure 2 described a puzzle set that consists of several task items. Each task item has two types of data: the scaffolding data and the simulation data.

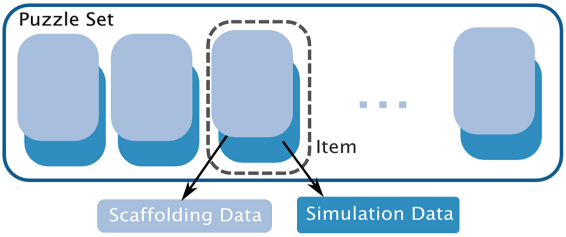


Fig. 2. A puzzle set and a task item.

Using the game format, a game consists of a sequence of inter-related tasks that can be easily created to learn problem solving skills. Some tasks can be recalled several times to promote a retrieval practice, which is essential for learning [5].

3.2 Knowledge Based Tutor

Beside the scaffolding data in the task item, we created QTut, a knowledge based tutor. QTut allows students to query some information in relation to the task at hand.

To support the extensibility of QTut, we created *knowledge triplet* (Qs, R, DA), where Qs refers to a list of query samples; R represents a response to a list of query samples Qs; and DA denotes dialog act (Program 1). The knowledge triplet (subsequently called triplet) represents QTUt knowledge on learning topics. Consequently, the number of triplets is contingent on the coverage of the learning topics in the game.

Table 2. An example of N-gram TF-IDF table with 2 triplets.

N-gram words	TF-IDF of triplet 1	TF-IDF of triplet 2
Net force	0.40	0
Normal force	0	0.4
Force	0.10	0.10

```
Program 1 (A knowledge triplet):
{  "Qs": ["Define normal force","What is normal  force"],
   "R": "Normal force  (N) is the component (perpendicular to the surface
         of contact) of the contact force exerted on an object by,
         for instance, the surface of a floor or wall, preventing the
         object from penetrating the surface",
   "DA": { "key": ["what", "define"],
           "intention": "ASK_EXPLAIN" }
}
```

Using NLTK, we use the triplets to construct a N-gram term frequency - inverse document frequency (TF-IDF) table (Table 2) that measures how concentrated the occurrence of a given word in a collection of triplets. Words with high TF-IDF numbers imply a strong relationship with the triplet they appear in, suggesting that if that word were to appear in a query, the triplet could be of interest to the student.

To match a user query to a triplet, we also transformed the user query into a set of N-gram words. We developed a Naive Bayes classifier to determine the similarity between the set of query words and the triplets using TF-IDF information. QTut subsequently ranks the similarity values in descending order and removes triplets that have similarity values below a certain threshold. QTut performs intention matching on the DA of the remaining triplets with the following rules: if it finds a match, then returns the corresponding triplet; otherwise, returns the triplet with the highest similarity value.

QTut has two response modes: "text" and "text-auditory". For text-auditory mode, we use a free text-to-speech (TTS) web service[2] to convert texts into speeches. The procedure is that QTut sends the texts to the TTS web API using HTTP GET and the TTS web API subsequently synthesizes the speeches and sends them to QTut. This supports both extensibility and scalability.

3.3 Modular System

To facilitate scalability and reusability, the game system is divided into functionality modules (Fig. 3): (a) the tutoring module delivers questions, provides hints and feedbacks, and responds to queries; (b) the physics simulation module

[2] VoiceRSS Text To Speech (http://voicerss.org/).

handles all graphical events based on laws in physics; (c) the delivery module draws a task item from the puzzle set either in random, sequential, or difficulty based order; and (d) the data module accesses, organizes, and manipulates game database (i.e., game contents, game configuration, and user log).

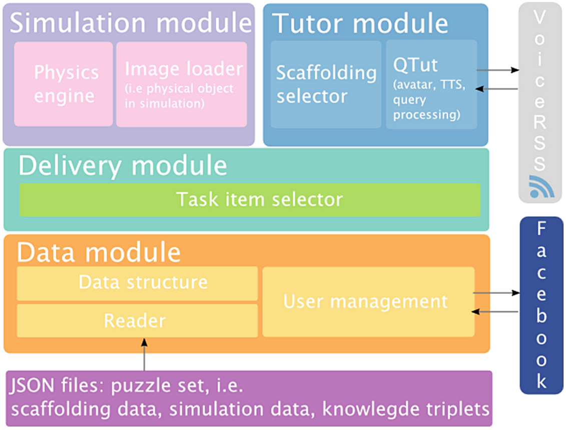


Fig. 3. A stack of modules as a complete system architecture.

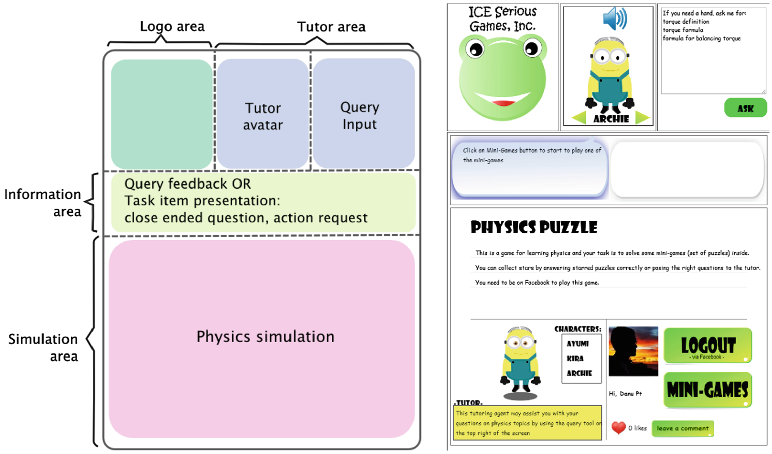
To minimize the needs of user management and to support the game distribution, the system is connected to a social networking platform (Facebook) using *Facebook Javascript API*[3]. The system extracts user information on Facebook to be stored into the database.

3.4 Graphical User Interface (GUI)

Good GUI is essential to improve goal clarity and sense of control of artefacts. To this end, the layout of the game GUI was designed using grid systems to group all elements according to their functionalities. This allows the game users to easily comprehend and navigate the interface [11]. Figure 4 shows the wireframe of the game GUI: tutor area on the top right consists of a tutor avatar and an input text to enter query for the tutor, information area on the middle presents feedback and task from the tutor, and the simulation area on the bottom plays physics events. The final GUI of the Physics game prototype is shown in Fig. 4(b).

The GUI elements (e.g., buttons and playable objects) use the feedforward and feedback concept to allow intuitive interaction. Feedforward is the information that occurs during or after user action, e.g., on-screen messages indicating what to do. Feedback is 'the return of information about the result of a process or activity' [12], e.g., clicking on a button opens a new window. Figure 5(a) shows the use of feedforward and feedback in a Logout button and Fig. 5(b) shows the

[3] Facebook Developer API (https://developers.facebook.com/).



(a) the wireframe of the game (b) the game GUI

Fig. 4. Wireframe of the game vs. the game GUI

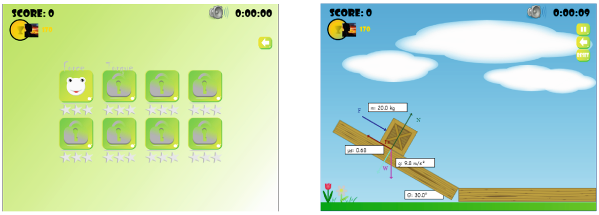
(a) Feedforward and feedback in a Logout button (b) Feedforward of a ruler in simulation

Fig. 5. Examples of feedforward and feedback.

feedforward for using a ruler in simulation. Feedforward conveys an implicit message that the logout button is click-able by changing its color upon mouse-over event; and the feedback responds to user action (a click) by changing the logout button into a login button. Feedforward is also used to help students in problem solving. For instance, a calculator button appears if a task item asks student to calculate force. The physics simulation shows a ruler if student needs to measure length or distance.

3.5 The Game Prototype

The game prototype has two levels: force and torque. The first level consists of nine close ended questions. The questions are either conceptual or procedural problems. The second level has six action requests that demands student to interact with objects in the simulation area. Figure 6(a) shows a list of game levels where all levels are locked except level 1 (force). Figure 6(b) shows a task item in the first level that asks about stationary state. Figure 6(c) shows a task item in the second level that demands student to balance the mobile toy. Each correct answer is awarded with ten points and a star-if the task item is a starred task item. A student passes a level if he earns two stars (three stars are available in each level) and scores above 50 %.



(a) Locked game levels (b) Game level 1, force (c) Game level 2, torque

Fig. 6. Game levels

4 Conclusions and Future Works

In this paper, we have presented our work on designing a physics game to support inquiry learning and retrieval practicing using simulation and knowledge based tutor (QTut). The game was implemented as an online puzzle game that used driving questions to encourage students to explore the simulation. We addressed three challenges in designing the system: extensibility, scalability, and reusability. Consequently, we defined game levels and game format to cope with extensibility. Also, knowledge triplets were designed to represent QTut knowledge. The system was divided into modules to allow scalability and reusability. The game GUI was designed using feedforward and feedback concepts on a grid system.

This work provided a baseline for creating educational games using the flow framework. We set the tasks with the increasing difficulties but we have not implemented difficulty adaptation into the system. It will be beneficial to explore the mechanism for difficulty adaptation. Task and user models proposed by [13] may fit to our case since our game prototype has leveled gameplay. In the future, we will test the game for its usefulness by assessing user performance and user perception of the games which is essential for serious games [14].

Acknowledgment. This work was supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA n 2010-0012. This work also is co-funded by the EU under the FP7, in the Games and Learning Alliance (GaLA) Network of Excellence, Grant Agreement nr. 258169.

References

1. Johnstone, A.H.: Why is science difficult to learn? things are seldom what they seem. Comput. Assist. Learn. **7**(2), 75–83 (1991)
2. Heyworth, R.: Procedural and conceptual knowledge of expert and novice students for the solving of a basic problem in chemistry. Int. J. Sci. Educ. **21**(2), 195–211 (1999)
3. Kirschner, P., Clark, R.: Work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educ. Psychol. **41**(2), 75–86 (2006)
4. Jong, T.D.: Technological advances in inquiry learning. Science **312**(5773), 532–533 (2006). (New York, N.Y.)
5. Karpicke, J.D., Blunt, J.R.: Retrieval practice produces more learning than elaborative studying with concept mapping. Science **331**(6018), 772–775 (2011). (New York, N.Y.)
6. Wouters, P., van Nimwegen, C., van Oostendorp, H., van der Spek, E.D.: A meta-analysis of the cognitive and motivational effects of serious games. J. Educ. Psychol. **105**(2), 249–265 (2013)
7. Kiili, K., de Freitas, S., Arnab, S., Lainema, T.: The design principles for flow experience in educational games. Procedia Comput. Sci. **15**, 78–91 (2012). 4th International Conference on Games and Virtual Worlds for Serious Applications(VS-GAMES 2012)
8. Kolb, D.A., et al.: Experiential Learning: Experience as the Source of Learning and Development, vol. 1. Prentice-Hall Englewood Cliffs, Upper Saddle River (1984)
9. Csikszentmihalyi, I.S.: Optimal Experience: Psychological Studies of Flow in Consciousness. Cambridge University Press, Cambridge (1992)
10. Pranantha, D., Bellotti, F., Berta, R., DeGloria, A.: A format of serious games for higher technology education topics: a case study in a digital electronic system course. In: International Conference on Advanced Learning Technologies, pp. 13–17, Rome. IEEE (2012)
11. Elam, K.: Grid Systems: Principles of Organizing Type (Design Briefs). Princeton Architectural Press, New York (2004)
12. Wensveen, S.A.G., Djajadiningrat, J.P., Overbeeke, C.J.: Interaction frogger: a design framework to couple action and function through feedback and feedforward. In: Designing Interactive Systems: Processes, Practices, Methods, and Techniques, pp. 177–184, New York, NY. ACM (2004)
13. Bellotti, F., Berta, R., De Gloria, A., Primavera, L.: Adaptive experience engine for serious games. IEEE Trans. Comput. Intell. AI Games **1**(4), 264–280 (2009)
14. Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., Berta, R.: Assessment in and of serious games: an overview. Adv. Hum. Comput. Interact. **2013**, 1 (2013)